Abstract

An apparatus to perform no-latency conditional branching has a sequencer (19) for executing program instructions including one or more conditional branch instructions. The conditional branch instruction is a binary word specifying a branch condition address and a conditional instruction. A branch unit (302,802) determines in hardware whether to branch according to the conditional instruction. The branch unit (302,802) has a programmable flag selection memory (502,902) and a plurality of first flag selectors (504,904). Each first flag selector (504,904) accepts a plurality of available flags (25,55) and selects a flag (508) based upon contents in the flag selection memory (502,904). A second flag selector (506,806) accepts the flags (508,908) from the first flag selectors (504,904) and selects one of the flags (508,908) to present as a branch flag (304,804) based upon the branch condition address (300). The branch flag (304) indicates to the sequencer (19) whether to branch to the destination address.

A method to compile a program to support no-latency conditional branching includes the steps of interpreting the source code and identifying each conditional branch instruction. For each conditional branch instruction, the compiler determines a set of flags upon which the conditional branch instruction is based. The compiler identifies a flag selection register value (702-718) for each flag and stores as flag selection register array elements. The compiler then assigns a branch condition address (300) for the conditional branch instruction and encodes it as a binary word. The compiler then stores the encoded one or more conditional branch instructions and the flag selection register array elements in an object code format.